Determinació del subgrup de 7-Sylow d'una corba el·líptica

Autor: Sergi Fragua Ortega

Director: Josep M. Miret Biosca

Universitat de Lleida
Escola Politècnica Superior
Enginyeria Tècnica en Informàtica de Sistemes

Treball de Fi de Carrera

Setembre de 2007

Índex

Índex de figures

Índex de taules

Capítol 1

Introducció

Les corbes el·líptiques van ser proposades per primer cop per ser utilitzades en aplicacions criptogràfiques en 1985 de forma independent per Miller[6] i Koblitz[5]. Les corbes el·líptiques en si mateixes porten sent estudiades durant segles i es troben entre els objectes més ricament estructurats i estudiats de la teoria de nombres. Entrarem a parlar més detingudament de les corbes el·líptiques en capítols posteriors, ara ens centrarem en explicar un dels camps on s'apliquen, parlarem doncs, de la criptografia.
Des de que l'home ha tingut la necessitat de comunicar-se amb els demés també l'ha tingut de que alguns dels seus missatges sol fossin coneguts per les persones a les quals anaven dirigits. La necessitat de poder enviar missatges de forma que només fossin entesos pels destinataris va fer que es creessin sistemes de xifrat, de forma que un missatge després d'un procés de transformació, el que anomenem xifrat, sol pugui ser llegit seguint un procés de desxifrat. Les civilitzacions més antigues (egípcia, mesopotàmica, xina ...) ja utilitzaven aquestos mètodes. Un dels primers mètodes d'encriptació que està documentat és atribuït a Juli Cèsar, que es basava en la substitució de les lletres d'un document per la tercera lletra que li correspongués en l'alfabet. Amb el temps i degut principalment al seu ús militar, els sistemes criptogràfics han anat avançant en complexitat, fins arribar als sistemes actuals on la informàtica està per tot arreu i la necessitat de seguretat s'ha convertit en un aspecte essencial. A l'actualitat a la vida real, estem acostumats a enviar o rebre cartes postals que venen tancades en un sobre per a que la seva lectura estigui reservada solament a nosaltres o al seu destinatari. Al món virtual, en el cas de l'email això no és així, ja que el que enviem és la carta sense el sobre, és a dir, sense cap protecció que pugui impedir la seva lectura per part de qualsevol que pugui interceptar-la.

Com funciona la criptografia? La paraula criptologia prové de les paraules gregues Kryto i logos i significa, estudi de l'ocult. Una branca de la criptologia és la criptografia, que s'ocupa del xifrat de missatges. Aquesta es basa en que l'emissor emet un missatge en clar, que és tractat mitjançant un xifrador amb l'ajuda d'una clau, per crear un text xifrat, ajudant-se d'una altra clau per tal d'obtenir el text original. Les dos claus implicades en el procés de xifrat/desxifrat poden ser o no igual depenen del sistema de xifrat emprat. Podem trobar tres sistemes

de xifrat, sistema simètric, sistema asimètric i sistema híbrid. Els sistemes de xifrat simètrics són aquells que utilitzen la mateixa clau per xifrar i desxifrar un document. El principal problema de seguretat resideix en l'intercanvi de claus entre l'emissor i el receptor ja que els dos han d'utilitzar la mateixa clau. Per lo tant s'ha de buscar també un canal de comunicació que sigui segur per l'intercanvi de la clau. És important que aquesta clau sigui molt difícil d'esbrinar ja que avui en dia els ordinadors poden obtenir claus molt ràpidament. Per exemple l'algoritme de xifrat DES usa una clau de 56 bits, el que significa que hi ha 72 bilions de claus possibles. Actualment ja existeixen ordinadors especialitzats que són capaços de provar totes aquestes possibles claus en qüestió d'hores. Avui en dia s'estan utilitzant ja claus de 128 bits que augmenten "*l'espectre*"de claus possibles (2 elevat a 128) de forma que encara que s'unissin tots els ordinadors existents en aquests moments no ho aconseguirien desxifrar en milers de milions d'anys.

Els sistemes de xifrat asimètrics són anomenats també sistemes de xifrat de clau pública. Aquest sistema de xifrat utilitza dos claus diferents, una és la clau pública i es pot enviar a qualsevol persona i l'altra que s'anomena clau privada, s'ha de guardar per a que ningú tingui accés a ella. Per enviar un missatge, el remitent utilitza la clau pública del destinatari per xifrar el missatge. Un cop l'ha xifrat, solament amb la clau privada del destinatari es pot desxifrar, ni tan sols qui ha xifrat el missatge pot tornar a desxifrar-lo. Per això es pot donar perfectament la clau pública per a que tothom qui vulgui comunicar-se amb el destinatari ho pugui fer. Dintre dels sistemes de xifrat asimètrics podem trobar el sistema RSA. L'algoritme RSA, va ser ideat al 1977 per Ron Rivest, Adi Shamir i Leonard Adleman (RSA). És el més usat a l'actualitat, senzill de comprendre i implementar, encara que la longitud de les seves claus és bastant considerable (ha passat dels seus 200 bits originals fins a 2048 actualment). S'utilitzen les avantatges proporcionades per les propietats dels nombres primers quan s'apliquen sobre ells operacions matemàtiques basades en la funció mòdul. La robustesa d'aquest sistema es basa en la gran dificultat que presenta el problema de la factorització d'enters (PFE), on l'enter és producte de dos primers grans. Encara que l'avanç tecnològic fa que cada dia sigui més ràpid un possible atac per força bruta, el simple fet d'augmentar la longitud de les claus emprades, suposa un increment de la càrrega computacional suficientment gran per a que aquest tipus d'atac sigui inviable.
Per altra banda, el sistema d'ElGammal, és un altre tipus d'esquema de clau pública, basat en el problema del logaritme discret. El problema del logaritme discret consisteix en, donat un grup cíclic $(G, *)$ finit de cardinal n, un generador g i un element h que pertany a G, obtenir k, tal que $g^k = h$.
Ahmed ElGammal va proposar un esquema basat en aquest problema sobre el grup multiplicatiu d'un cos finit. Podem dir, que el sistema d'ElGammal és equivalent en quant a eficàcia, al sistema RSA, però aquest últim ha desenvolupat una variant millorada proposada per Miller[6] i Koblitz[5], basada en una versió el·líptica del problema del logaritme discret.
La criptografia basada en corbes el·líptiques (CCE), basa la seva seguretat amb el Problema del Logaritme Discret El·líptic (PLDE), això vol dir que donats P

i Q, punts d'una corba el·líptica E sobre un cos $\mathbb{F}_p$ tals que $Q \in < P >$, s'ha de trobar un nombre enter x tal que $xP = Q$. Pel que fa a la implementació, hi ha bons algoritmes que sumen punts racionals.
Per tant els avantatges que ofereixen els sistemes CCE respecte els RSA són, entre d'altres la longitud de les claus. Es pot veure que mentre amb RSA s'ha de fer ús d'una clau de 1024 per oferir una seguretat considerable, la CCE sol utilitza 163 bits per oferir la mateixa seguretat, de la mateixa manera les claus RSA de 2048 són equivalents en seguretat a 210 de CCE. Això és degut a que per resoldre el PLDE l'únic algoritme conegut, pren un temps d'execució totalment exponencial, mentre que el que resol PFE pren un temps subexponencial. Una altra avantatge de la CCE és que es pot treballar en cossos finits de característica 2. En aquest cas és possible construir una aritmètica que optimitzi la rapidesa i construir un circuit especial per aquesta aritmètica, i que es coneix com Base Normal Óptima. Això, fa que la CCE sigui ideal per a ser implementada quan el poder de comput i l'espai del circuit és reduït, on sigui requerida una alta velocitat de processament o grans volums de transaccions, on l'espai d'emmagatzemament, la memòria o l'ample de banda sigui limitat. Això permet el seu ús a Smart Cards, Telèfons Mòbils, Fax, PCs ...
La CCE és doncs, la millor candidata per reemplaçar a les aplicacions que tenen implementat RSA, doncs defineixen també esquemes de firma digital, intercanvi de claus simètriques i d'altres.

En aquest projecte de final de carrera, el primer que farem serà introduir els conceptes matemàtics i de corbes el·líptiques bàsics, necessaris per a poder entendre millor els capítols posteriors on entrarem a parlar detalladament de les corbes el·líptiques i les seves característiques. Un cop explicat aquests conceptes, s'explicarà com determinar el subgrup de 7-Sylow d'una corba el·líptica definida sobre un cos finit i realitzarem la implementació de l'algorisme que utilitzarem per determinar-lo. El subgrup de 7-Sylow ens donarà informació parcial sobre el cardinal de la corba, i gràcies a això, podrem tindre un criteri per tal de rebutjar corbes el·líptiques criptogràficament no vàlides en el cas que vulguem treballar amb criptosistemes basats en el problema del logaritme discret. Finalment, es mostraran els resultats obtinguts a partir de diverses proves realitzades amb les nostres implementacions i acabarem explicant les conclusions a les que hem arribat després de realitzar aquest treball.
Els aspectes que tractem en aquest treball sobre els subgrups de ℓ-Sylow d'una corba el·líptica, amb $\ell = 7$, ja han estat estudiats anteriorment considerant d'altres ℓ's en els TFC's [1], [8], [10] ,[11] i en casos més generals a [7]. La font original on es pot trobar l'estudi complet és la tesi de Ramiro Moreno [9]. Tots ells m'han set de gran ajuda a l'hora de realitzar el meu treball.

Agraïments

Agraeixo especialment a Javier Valera l'ajut i la paciència que ha tingut a l'hora de respondre i explicar-me els meus dubtes i a Josep M. Miret la seva ajuda i el seu esforç, a l'hora de resoldre'm els múltiples dubtes i problemes que m'han anat sorgint en el progrés de realització d'aquest treball.

Capítol 2

Preliminars matemàtics

En aquest capítol s'explicaran els conceptes que hem de tindre clars per poder entendre la teoria de corbes el·líptiques. La majoria d'aquests conceptes ja han estat estudiats durant la carrera, concretament a l'assignatura d'Àlgebra [3], però és convenient recordar-los, doncs ens seran útils a l'hora de veure els posteriors capítols.

2.1 Teoria de Grups

2.1.1 Lleis de composició interna

Una *llei de composició interna* o *operació interna* sobre un conjunt $\mathcal{C}$, és una aplicació del producte cartesià $\mathcal{C} \times \mathcal{C}$ en $\mathcal{C}$.
Si $f : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ és una llei de composició interna i $a, b \in \mathcal{C}$, aleshores a l'element $f(a, b)$ de $\mathcal{C}$ se'l pot representar de diferents formes: $a \oplus b$, $a \otimes b$, $a * b$, etc. Direm que $(\mathcal{C}, \oplus)$ és un conjunt $\mathcal{C}$ amb una operació interna $\oplus$.

Característiques de les lleis de composició interna

Donat $(\mathcal{C}, \oplus)$, l'operació interna $\oplus$ pot complir les següents propietats :

- *Associativa*, si $a \oplus (b \oplus c) = (a \oplus b) \oplus c, \ \forall a, b, c \in \mathcal{C}$;
- *Commutativa*, si $a \oplus b = b \oplus a, \ \forall a, b \in \mathcal{C}$;
- *Existència d'element neutre (és unic)* e, si $e \in \mathcal{C}$ i $a \oplus e = e \oplus a = a, \ \forall a \in \mathcal{C}$;
- *Existència d'element simètric* o *invertible* de a en $\mathcal{C}$, ve representat per a^{-1}, si $\exists a^{-1} \in \mathcal{C}$ tal que $a \oplus a^{-1} = a^{-1} \oplus a = e$, on e es l'element neutre de $(\mathcal{C}, \oplus)$.

Donat $(\mathcal{C}, \oplus, \otimes)$, és diu que l'operació interna $\otimes$ és *distributiva* respecte de l'operació interna $\oplus$, si satisfà:

$$a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c) \quad \text{i} \quad (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c), \quad \forall a, b, c \in \mathcal{C}.$$

2.1.2 Grups

Un conjunt $\mathcal{G}$ dotat d'una operació interna $\oplus$ és un grup, si l'operació interna $\oplus$ en $\mathcal{G}$ satisfà les propietats següents:

- Propietat Associativa;
- Té element neutre;
- Tot element de $\mathcal{G}$ té simètric.

Si a més se satisfà la propietat commutativa, es diu que $(\mathcal{G}, \oplus)$ és un *grup abelià*.

Grups finits
Es diu que un grup $\mathcal{G}$ és finit si té un nombre finit d'elements. L'*ordre (cardinal) d'un grup finit* $\mathcal{G}$, denotat per $\#\mathcal{G}$, és el nombre d'elements que té el grup.
L'*ordre d'un element* a de $\mathcal{G}$, denotat per $\#a$, és el menor $n \in \mathbb{N}$ tal que $a^n = a \oplus \ldots^n \ldots \oplus a = e$, on e és l'element neutre de $(\mathcal{G}, \oplus)$.

Congruències
Donat un enter $n \neq 0$, es diu que dos nombres enters a i b son *congruents* mòdul n si $a - b$ és un múltiple de n, i s'escriu :

$$a \equiv b \ (mod\ n)$$

En les congruències trobem les següents equivalències :

- $a \equiv b \ (mod\ n)$;
- La resta de dividir a per n és la mateixa que la resta de dividir b per n;
- L'enter b és de la forma $a + k \cdot n$, on $k \in \mathbb{Z}$.

La relació de congruència mòdul n definida en $\mathbb{Z}$ és una relació d'equivalència.

El grup $(\mathbb{Z}_n, +)$
El conjunt quocient de $\mathbb{Z}$ per la relació de congruència mòdul n, representat per $\mathbb{Z}_n$, està format per totes les classes de congruència diferents mòdul n, és a dir,

$$\mathbb{Z}_n = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\}, \text{ on } \overline{a} = \{a + k \cdot n \mid k \in \mathbb{Z}\}.$$

El conjunt $\mathbb{Z}_n$ dotat de l'operació interna $+$ definida per

$$\overline{a} + \overline{b} = \overline{a + b},$$

té estructura de grup abelià finit d'ordre n.

Homomorfisme i isomorfisme
Una aplicació f d'un grup $(\mathcal{G}, \oplus)$ a un altre grup $(\mathcal{G}', \otimes)$ es diu *homomorfisme* de $\mathcal{G}$ en $\mathcal{G}'$ si, per a qualsevol $x, y \in \mathcal{G}$ es verifica:

$$f(x \oplus y) = f(x) \otimes f(y).$$

Si l'aplicació f és exhaustiva s'anomena epimorfisme, si l'aplicació f és injectiva llavors f s'anomena monomorfisme. Si un homomorfisme és a la vegada injectiu i exhaustiu, llavors s'anomena **isomorfisme**.
Es diu que un grup $\mathcal{G}$ és isomorf a un grup $\mathcal{G}'$ quan es pot trobar un isomorfisme entre ambdós.

Coeficients de torsió d'un grup abelià finit
Un grup abelià finit $\mathcal{G}$ és isomorf a un producte directe $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \ldots \times \mathbb{Z}_{n_r}$, on n_{i+1} divideix a n_i, $\forall i$ tal que $1 \leq i < r$. S'anomenen *coeficients de torsió de* $\mathcal{G}$ als nombres n_i i *rang de* $\mathcal{G}$ a l'enter r. Si $r = 1$, $\mathcal{G}$ és cíclic.

2.2 Anells i Cossos

2.2.1 Anells

Un conjunt $\mathcal{A}$ dotat de dues operacions internes $(\oplus, \otimes)$, té estructura d'*anell*, és a dir, $(\mathcal{A}, \oplus, \otimes)$ és un anell, si:

- $(\mathcal{A}, \oplus)$ és un grup abelià;
- L'operació interna $\otimes$ és associativa;
- L'operació interna $\otimes$ és distributiva respecte de l'operació $\oplus$.

Si l'anell $(\mathcal{A}, \oplus, \otimes)$ té element neutre, anomenat *unitat*, respecte de l'operació interna $\otimes$, es diu que l'anell és unitari.
Si l'operació interna $\otimes$ satisfà la propietat commutativa, es diu que l'anell $(\mathcal{A}, \oplus, \otimes)$ és un anell commutatiu.
Es diu que $a \in \mathcal{A} - \{0\}$ és un *divisor de zero* d'un anell $(\mathcal{A}, \oplus, \otimes)$, on 0 és l'element neutre de $(\mathcal{A}, \oplus)$, si existeix un element $b \in \mathcal{A} - \{0\}$ tal que $a \otimes b = 0$ o $b \otimes a = 0$. Un anell commutatiu unitari sense divisors de zero s'anomena *domini d'integritat*.

L'anell $(\mathbb{Z}_n, +, \cdot)$
El grup abelià finit $(\mathbb{Z}_n, +)$ dotat de l'operació interna $\cdot$ definida per :

$$\overline{a} \cdot \overline{b} = \overline{a \cdot b},$$

té estructura d'anell unitari i commutatiu amb divisors de zero si n no és primer. Al conjunt d'elements invertibles de $\mathbb{Z}_n$ es representa per $\mathbb{Z}_n^*$. Aquest conjunt amb l'operació $\cdot$ té estructura de grup abelià.

2.2.2 Cossos

Un conjunt $\mathbb{K}$ dotat de dues operacions internes $(\oplus, \otimes)$ és un *cos* si :

- $(\mathbb{K}, \oplus, \otimes)$ és un anell unitari;

- Tot element de $\mathbb{K}$ diferent del 0, on 0 és l'element neutre de $(\mathbb{K}, \oplus)$, és invertible, és a dir,

$$\forall a \in \mathbb{K}^* = \mathbb{K} - \{0\} \; \exists a^{-1} \in \mathbb{K} \text{ tal que } a \otimes a^{-1} = a^{-1} \otimes a = 1,$$

  on 1 és l'element neutre de $(\mathbb{K}, \otimes)$, anomenat *unitat*. L'element a^{-1} es diu que és l'*invers* de a.

Si, a més, l'operació interna $\otimes$ satisfà la propietat commutativa, es diu que $(\mathbb{K}, \oplus, \otimes)$ és un *cos commutatiu*.
El conjunt $\mathbb{K}^*$ té estructura de grup respecte l'operació interna $\otimes$.

Cossos finits
Es diu que un cos $\mathbb{F}$ és finit si té un nombre finit d'elements. L'*ordre* d'un cos finit $\mathbb{F}$, que es representa per $|\mathbb{F}|$, és el nombre d'elements que té el cos i sempre és de la forma $q = p^m$, on p és un nombre primer (*característica* del cos) i m és un nombre natural més gran o igual que 1 (*grau de l'extensió* del cos). Si $m = 1$, l'anell $(\mathbb{Z}_p, +, \cdot)$ és un cos finit de p elements que es denota $\mathbb{F}_p$.

Capítol 3

Corbes el·líptiques

En aquest capítol explicarem la teoria de les corbes el·líptiques que tant important és en el camp de la criptografia i de la qual fins ara, només havíem parlat de forma molt breu en comptades ocasions. Concretament, parlarem sobre les corbes el·líptiques definides sobre un cos $\mathbb{F}_p$. D'aquesta manera ens resultarà més senzill entendre aquest treball i el capítol següent en el qual entrarem a parlar de l'algoritme que hem desenvolupat.

3.1 Introducció a les corbes el·líptiques

Una corba el·líptica sobre un cos $\mathbb{K}$, és una corba projectiva de gènere 1 la qual ha de tindre almenys un punt racional. Aquesta darrera condició no és necessària si $\mathbb{K}$ és algebraicament tancat, però és decisiva en el cas contrari. L'existència d'un punt racional ens permet dotar a la corba d'una estructura de grup. Per aquest motiu, en la definició de corba el·líptica no solament és necessari que hi hagi un punt racional, sinó que també hem de seleccionar un d'aquests punts per a que representi el paper d'element neutre. Una corba el·líptica sobre un cos $\mathbb{K}$ es defineix com a un parell $(E, \mathcal{O})$, on E és una corba projectiva de gènere 1 i $\mathcal{O} \in E(\mathbb{K})$, o també com a un conjunt de punts $(x, y) \in \mathbb{K} \times \mathbb{K}$ que satisfan l'equació general de Weierstrass que s'indica a continuació :

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

on les a_i són elements de $\mathbb{K}$. Per evitar que la corba tingui punts singulars, el discriminant de la corba ha de ser diferent de 0. Si utilitzem un canvi lineal de variables obtindrem, sempre que la característica del cos sigui diferent de 2 i de 3, una equació de la forma :

$$y^2 = x^3 + Ax + B,$$

on x i y són indeterminades i $A,\ B \in \mathbb{K}$. Aquesta equació que anomenarem $E_{A,B}/\mathbb{K}$, més simplificada que la anterior, rep el nom de forma reduïda de Weierstrass i per a que tingui estructura de corba el·líptica el discriminant $\Delta = 4A^3 + 27B^2$ ha de ser diferent de 0. Llavors, podem definir el conjunt de punts d'una corba el·líptica $E_{A,B}/\mathbb{K}$ com:

$$E_{A,B}(\mathbb{K}) = \{(x, y) \in \mathbb{K} \times \mathbb{K} \mid y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}_{E_{A,B}}\}$$

on $\mathcal{O}_{E_{A,B}}$ és l'anomenat punt de l'infinit de la corba que permet dotar a aquest conjunt d'estructura de grup abelià. Tota corba el·líptica és isomorfa a una corba el·líptica plana, i encara més a una corba determinada per una equació particular com és l'equació de Weierstrass. Tota equació de Weierstrass defineix una corba projectiva plana amb l'únic punt a l'infinit, $\mathcal{O} = [0, 1, 0]$. En principi aquesta corba no tindria perquè ser necessàriament el·líptica, ja que pot tindre punts singulars, però si això no es dóna, podem dir que la corba és el·líptica, ja que tota cúbica no singular té génere 1. Tota corba el·líptica admet una equació de Weierstrass, com a conseqüència del teorema de Riemann-Roch.

3.2 Suma de punts en una corba el·líptica

Al conjunt de punts d'una corba el·líptica $E_{A,B}/\mathbb{K}$ podem definir-hi una operació suma. Aquesta operació es pot expressar gràficament pel mètode de la corda i la tangent, però també, es pot expressar de forma algebraica, tal i com explicarem en aquest apartat.

3.2.1 Mètode de la corda i la tangent

Per a la suma de dos punts de una corba el·líptica s'utilitza el **mètode de la corda i la tangent** el qual consisteix en traçar una línia recta que passi pels dos punts que volem sumar, els quals anomenarem P i Q. L'equació de la corba és de grau 3 i la línia de grau 1, per tant existeixen sempre tres solucions, en aquest cas la tercera solució apareix a la figura com a $-R$. Un cop tenim aquesta tercera solució (punt), tracem una recta paral·lela a l'eix Y que passi per aquest punt $-R$ de manera que tornarà a tallar la corba en un altre punt simètric a $-R$ el qual anomenarem R i aquest serà el punt suma que volíem obtenir, $R = P+Q$.

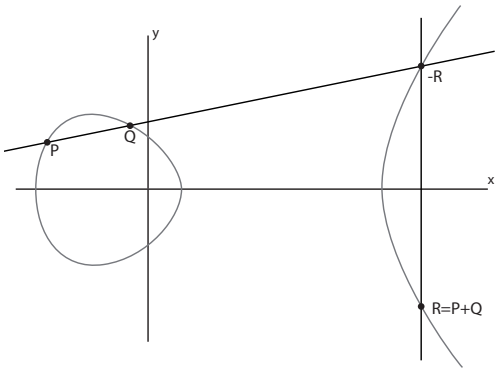

Figura 3.1: Suma de dos punts diferents pel mètode de la corda i la tangent.

Si P i Q són iguals, tracem la recta tangent a la corba respecte el punt P tal

com hem fet en el cas anterior. Aquesta recta talla la corba en un altre punt que anomenem $-R$, i un cop ja tenim aquest punt, traçant una recta paral·lela a l'eix Y que passi per $-R$ obtindrem el seu simètric R que tal i com passava en el cas en que els punts eren diferents, serà el punt suma que anomenem S. Per tant, $S = 2P$.

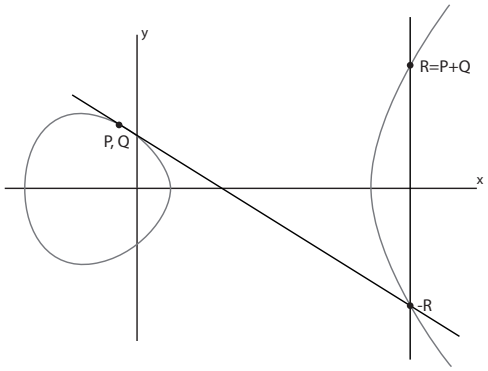

Figura 3.2: Suma de dos punts iguals pel mètode de la corda i la tangent.

Amb aquesta operació, $(E_{A,B}(\mathbb{K}), +)$ té estructura de grup abelià amb punt de l'infinit $\mathcal{O}_{E_{A,B}}$, com a punt neutre.

3.2.2 Càlcul algebraic de la suma de dos punts

Donada una corba el·líptica $E_{A,B}/\mathbb{K}$ i donats els punts $P = (x_1, y_1)$ i $Q = (x_2, y_2) \in E_{A,B}(\mathbb{K})$, no resulta complicat obtenir fórmules per calcular les coordenades del punt $P+Q = (x_3, y_3)$ a partir de les coordenades del punt $P = (x_1, y_1)$ i del punt $Q = (x_2, y_2)$.
Realitzant una sèrie de càlculs, obtenim que les coordenades del punt suma, es representen amb les següents equacions les quals, depenen de si els punts P i Q són diferents o no, ja que la pendent α de la recta PQ és diferent per cada cas, tenim doncs :

$$\begin{aligned} x_3 &= \alpha^2 - x_1 - x_2, \\ y_3 &= -y_1 + \alpha(x_1 - x_2). \end{aligned}$$

Si $P \neq Q$ tenim que la pendent α de la recta tangent en P és :

$$\alpha = \frac{y_1 - y_2}{x_1 - x_2}$$

En canvi si $P = Q$ tenim que la pendent α de la recta tangent en P és en aquest cas :

$$\alpha = \frac{3x_1^2 + A}{2y_1}$$

Si substituïm aquestos valors de α a les equacions de x_3 i y_3 obtindrem les coordenades del punt suma.

3.2.3 Múltiples d'un punt

A partir de l'operació de suma, es pot definir l'operació producte d'un punt P per un enter n amb la següent expressió:

$$nP = \begin{cases} \underbrace{P + P + ... + P}_{n \text{ vegades}} & \text{si } n > 0, \\ \mathcal{O}_{E_{A,B}} & \text{si } n = 0, \\ \underbrace{(-P) + (-P) + ... + (-P)}_{n \text{ vegades}} & \text{si } n < 0. \end{cases}$$

Però apart d'aquest procediment per calcular nP existeix un altre algorisme que empra la pròpia llibreria LiDIA i que resulta molt més eficient que l'anterior que rep el nom d'**algoritme del camperol rus**. Nosaltres utilitzarem l'algoritme del camperol rus, ja que ens permet fer el càlcul nP amb un menor cost.

Algoritme del camperol rus

És un algoritme d'exponenciació ràpida que ens permet calcular nP. Consisteix en sumar i multiplicar el punt P amb potències de dos. Veiem-ho amb un exemple:

$$\begin{aligned} n &= 15_{10} = 1111_2 \\ nP &= (2^3 + 2^2 + 2^1 + 2^0)P \\ nP &= (2(2^2 + 2 + 1)P + P) \\ nP &= (2(2(2 + 1)P + P) + P) \\ nP &= (2(2(2P + P) + P) + P) \end{aligned}$$

Com es pot veure el cost del càlcul és de l'ordre $\mathrm{O}(\log_2 n)$, un cost molt menor, al que tenim si ho fem amb sumes successives, que és d'ordre $\mathrm{O}(n)$.

3.3 Corbes el·líptiques sobre cossos $\mathbb{F}_p$

Quan tenim una corba el·líptica definida mitjançant una equació amb coeficients enters podem prendre congruències mòdul un primer per poder passar a una equació definida en un cos finit. Si el primer no divideix al discriminant de la corba, el resultat serà novament una corba el·líptica.

Una corba el·líptica sobre un cos finit $\mathbb{F}_p$, on p és un nombre primer, ve definida per una equació de la forma:

$$y^2 \equiv x^3 + Ax + B \pmod{p},$$

on A, B són elements de $\mathbb{F}_p$ i $4A^3 + 27B^2 \not\equiv 0 \pmod{p}$. El conjunt de punts de la corba són els punts (x, y) tals que x i $y \in \mathbb{F}_p$ i que satisfan l'equació de la corba. Totes les propietats vistes anteriorment sobre cossos també són aplicables a les corbes el·líptiques sobre cossos $\mathbb{F}_p$, però aquestes tenen una sèrie de propietats i característiques pròpies que explicarem a continuació.

3.3.1 Cardinal

El cardinal d'una corba el·líptica $E_{A,B}/\mathbb{F}_p$, que denotem per $\#E_{A,B}(\mathbb{F}_p)$, és el nombre de punts que conté la corba amb coordenades a $\mathbb{F}_p$, més el punt de l'infinit $\mathcal{O}_{E_{A,B}}$. Trobar aquest nombre de punts resulta un problema computacionalment difícil de resoldre.

En general, una corba el·líptica $E_{A,B}/\mathbb{F}_p$ definida sobre un cos $\mathbb{F}_p$, no tindria perquè tindre punts racionals a part del punt a l'infinit $\mathcal{O}_{E_{A,B}}$. Malgrat això, està comprovat que si $\mathbb{F}_p$ és un cos finit amb mòdul més gran que 4, $E_{A,B}/\mathbb{F}_p$ té almenys un punt racional no trivial.
A un punt que satisfà l'equació general de Weierstrass se l'anomena punt racional. Si el cos és finit, llavors el conjunt de punts (x, y) que satisfan l'equació és finit i se l'anomena conjunt de punts racionals de la corba $E_{A,B}$ sobre el cos $\mathbb{F}_p$. Al conjunt de punts racionals el podem representar com :

$$E_{A,B} : \mathcal{O}, P_1, P_2, P_3, P_4, ..., P_n.$$

$E_{A,B}$ representa l'equació i $\mathcal{O}$ és un punt que fa el paper de 0 (punt a l'infinit).

Exemple: Si agafem una corba el·líptica simple, d'equació $y^2 = x^3 + 4x + 3$ i el cos és $\mathbb{F}_5$, llavors, les parelles que satisfan aquesta equació són {(2,2),(2,3)}, per tant el conjunt de punts de la corba el·líptica és $E(\mathbb{F}_5)$={$\mathcal{O}$,(2,2),(2,3)}, és a dir que $E/\mathbb{F}_5$ té 3 punts racionals.
El teorema de Hasse acota el cardinal d'una corba el·líptica.

Teorema 1 *(Hasse) Sigui $E_{A,B}$ una corba el·líptica definida sobre un cos finit $\mathbb{F}_p$, i sigui $m = \#E_{A,B}(\mathbb{F}_p)$, llavors:*

$$p + 1 - 2\sqrt{p} \leq m \leq p + 1 + 2\sqrt{p}.$$

De la mateixa manera, si $P \in E_{A,B}(\mathbb{F}_p)$, l'ordre d'aquest punt és l'enter positiu més petit n tal que $nP = \mathcal{O}_{E_{A,B}}$. Tenint en compte el teorema de Lagrange i les propietats dels grups, se satisfà:

$$\#P \mid \#E_{A,B}(\mathbb{F}_p).$$

Per a calcular el cardinal d'una corba el·líptica $E_{A,B}/\mathbb{F}_p$ podem utilitzar el mètode del recompte exhaustiu. Aquest mètode consisteix en substituir cada element de $\mathbb{F}_p$ a l'expressió $x^2 + Ax + B$ de la corba i realitzar el sumatori de tots els símbols de Legendre associats als elements de $\mathbb{F}_p$ obtinguts de la substitució. Més concretament,

$$\#E_{A,B}(\mathbb{F}_p) = 1 + \sum_{x \in \mathbb{F}_p} (1 + \chi(x^3 + Ax + B)).$$

El càlcul del cardinal pel mètode del recompte exhaustiu és ineficient, és per això que s'utilitzen altres algoritmes com per exemple el de Shanks-Mestre[13] o el de Schoof[12], que fan aquest càlcul de forma molt més efectiva.

3.4 Polinomis de divisió

Els polinomis de divisió ens son útils a l'hora de trobar els punts d'ordre n d'una corba $E/\mathbb{F}_p$, i les arrels d'aquests polinomis. Els polinomis de divisió ens donen les abscisses d'aquests punts.
Donada una corba el·líptica $E/\mathbb{F}_p$ d'equació $y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6$, es defineixen els polinomis divisors $\Psi_n(x,y)$ tal que $n \in \mathbb{Z}_{\geq 0}$ amb les expressions recursives següents:

$$\begin{aligned}
\Psi_0(x,y) &= 0 \\
\Psi_1(x,y) &= 1 \\
\Psi_2(x,y) &= 2y + a_1x + a_3 \\
\Psi_3(x,y) &= 3x^4 + b_2x^3 + 3b_4x^2 + 3b_6x + b_8 \\
\Psi_4(x,y) &= (2x_6 + b_2x^5 + 5b_4x^4 + 10b_6x^3 + 10b_8x^2 \\
&\quad +(b_2b_8 - b_4b_6)x + b_4b_8 - b_6^2)\Psi_2 \\
\Psi_{2n+1}(x,y) &= \Psi_{n+2}\Psi_n^3 - \Psi_{n-1}\Psi_{n+1}^3, si\ n \geq 2 \\
\Psi_{2n}(x,y) &= \frac{\Psi_{n+2}\Psi_{n-1}^2 - \Psi_{n-2}\Psi_{n-1}^2}{\Psi_2}, si\ n > 2
\end{aligned}$$

On les $b's$ venen donades per les següents expressions:

$$\begin{aligned}
b_2 &= a_1^2 + 4a_2 \\
b_4 &= 2a_4 + a_1a_3 \\
b_6 &= a_3^2 + 4a_6 \\
b_8 &= a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2
\end{aligned}$$

A partir del polinomi de divisió $\Psi_n(x,y) \in \mathbb{F}_p[x,y]$, podem definir el polinomi $f_n(x) \in \mathbb{F}_p[x]$ de la següent manera:

$$f_n(x) = \begin{cases} \Psi_n(x) & \text{si } n \text{ és senar,} \\ \dfrac{\Psi_n(x)y}{\Psi_2(x,y)} & \text{si } n \text{ és parell.} \end{cases}$$

Notem que quan n és senar, el polinomi $\Psi_4(x,y)$ no depèn de y i és per això que escriurem $\Psi_n(x)$. En canvi, quan n és parell, $\Psi_4(x,y)$, és de la forma $\Psi_n(x)y$.

Sigui $P = (x,y)$ un punt d'ordre n de la corba $E/\mathbb{F}_p$, aleshores, $f_n(x) = 0$. Per tant per trobar les abscisses d'un punt d'ordre n buscarem les arrels del polinomi $f_n(x)$.

Capítol 4

Determinació del subgrup de 7-Sylow

En aquest capítol, estudiarem com determinar el subgrup de 7-Sylow d'una corba el·líptica E, és a dir, el subgrup de punts d'ordre potència de 7.
Primer parlarem de les corbes el·líptiques amb punts d'ordre 7 i com podem obtenir aquestos punts. Seguidament, parlarem dels punts setena part i de com hem fet per trobar els diferents valors que ens interessen per tal d'obtindre el subgrup de 7-Sylow, diferenciant dos casos, el cas cíclic i el cas no cíclic. Per acabar, mostrarem i explicarem de la forma més clara possible, l'algorisme que hem fet servir per tal de determinar el subgrup.

4.1 Corbes el·líptiques amb punts d'ordre 7

Partint d'una corba E, sobre un cos finit $\mathbb{F}_p$, podem definir el subgrup de n-torsió de la corba, és a dir el grup de punts d'ordre n, junt amb el punt a l'infinit $\mathcal{O}_E$ que pertanyen a $E(\mathbb{F}_p)$ de la següent manera:

$$E[n](\mathbb{F}_p) = \{P \in E(\mathbb{F}_p) \mid nP = \mathcal{O}_E\}.$$

Aquestes expressions tracten el cas general, al meu treball parlarem en concret del cas en el que la n és 7 o més en general, una potència de 7.

Començarem doncs per calcular la 7-torsió de la corba, és a dir, els punts d'ordre 7. Si suposem que la corba té almenys un punt d'ordre 7, fent una translació d'aquest punt a l'origen, es pot escriure de la forma següent:

$$y^2 + (1 + c - c^2)xy + c^2(1 - c)y = x^3 + c^2(1 - c)x^2.$$

Com es pot veure, la nostra equació depèn únicament del paràmetre c.

Per poder trobar tots els punts d'una corba $E(\mathbb{F}_p)$, és indispensable que fem ús d'un polinomi de divisió en concret, que en el nostre cas és el **polinomi de 7-divisió**. Per l'obtenció d'aquest polinomi divisor, farem ús de les expressions recursives ja explicades a la secció 3.4.

Per tant si volem obtenir el polinomi Ψ_7 hem de utilitzar la expressió $\Psi_{2n+1}(x,y)$ amb $n=3$ quedant de la següent forma :

$$\Psi_7 = \Psi_{2*3+1} = \Psi_5\Psi_3^3 - \Psi_2\Psi_4^3$$

Si ho fem d'aquesta manera i substituïm les Ψ per les seves expressions, ens trobem amb l'inconvenient de que Ψ_2 conté una variable y, i a nosaltres ens interessa que tot estigui en funció de x, per tant mirarem de solucionar-ho de la següent forma:

Considerarem $\Psi_4 = \Psi\Psi_2$ llavors,

$$\Psi_7 = \Psi_{2*3+1} = \Psi_5\Psi_3^3 - \Psi_2\Psi_4^3$$

Si calculem Ψ_5 tenim:

$$\Psi_5 = \Psi_{2*2+1} = \Psi_4\Psi_2^3 - \Psi_1\Psi_3^3 = \Psi\Psi_2\Psi_2^3 - \Psi_1\Psi_3^3 = \Psi(\Psi_2^2)^2 - \Psi_3^3$$

Substituint a Ψ_7 ens queda:

$$\Psi_7 = \Psi_{2*3+1} = \Psi_5\Psi_3^3 - \Psi_2\Psi_4^3 = (\Psi(\Psi_2^2)^2 - \Psi_3^3)\Psi_3^3 - \Psi_2(\Psi\Psi_2)^3 = \Psi(\Psi_2^2)^2\Psi_3^3 - (\Psi_3^3)^2 - (\Psi_2^2)^2\Psi^3 = \Psi(\Psi_2^2)^2(\Psi_3^3 - \Psi^2) - (\Psi_3^3)^2$$

I substituint finalment, les Ψ's per les seves expressions obtindrem el nostre polinomi de 7-divisió per una corba d'equació en forma general de Weierstrass :

$$\Psi_7(x) = (2x^6 + b_2x^5 + 5b_4x^4 + 10b_6x^3 + 10b_8x^2 + (b_2b_8 - b_4b_6)x + b_4b_8 - b_6^2)((4x^3 + b_2x^2 + 2b_4x + b_6)^2)((3x^4 + b_2x^3 + 3b_4x^2 + 3b_6x + b_8)^3 - (2x^6 + b_2x^5 + 5b_4x^4 + 10b_6x^3 + 10b_8x^2 + (b_2b_8 - b_4b_6)x + b_4b_8 - b_6^2)^2) - ((3x^4 + b_2x^3 + 3b_4x^2 + 3b_6x + b_8)^3)^2$$

On les expressions que defineixen les $b's$ ja han estat introduïdes a l'apartat (3.4) i les $a's$ en termes de la c són les següents:

$$\begin{aligned} a_1 &= 1 + c + c^2 \\ a_2 &= c^2(1-c) \\ a_3 &= c^2(1-c) \end{aligned}$$

Aquest polinomi tindrà 3 solucions si ens trobem davant del cas cíclic o 24 solucions si es tracta del cas que el grup de 7-torsió sigui no cíclic. Aquestes solucions són les abscisses dels punts d'ordre 7. Com que cada abscissa té dos ordenades, tenim 6 o bé 48 punts d'ordre 7. Per tant, afegint el punt de l'infinit, el subgrup de 7-torsió és isomorf a $\mathbb{Z}_7$ o $\mathbb{Z}_7 \times \mathbb{Z}_7$, respectivament.

4.2 Subgrup de 7-Sylow d'una corba el·líptica

Quan parlem d'un subgrup de 7-Sylow d'una corba el·líptica $E(\mathbb{F}_p)$, ens estem referint a tot el conjunt de punts d'ordre una potencia de 7 que conté aquesta corba, i ho denotarem com $S_7(E(\mathbb{F}_p))$.
D'aquest subgrup podem dir que és de la forma:

$$S_7(E(\mathbb{F}_p)) \cong \mathbb{Z}_{7^n} \times \mathbb{Z}_{7^r}, \text{ on } n \geq r \geq 0$$

En el cas que $r = 0$ el subgrup és cíclic, mentres que si $n > 0$ es tracta del cas no cíclic.

Un altre aspecte important a tindre en compte és la valoració 7-àdica de $p - 1$ que ens permetrà estudiar una propietat de l'estructura $S_7(E(\mathbb{F}_p))$. La valoració 7-àdica expressada com $v_7(p-1)$ ve definida de la forma,

$$v_7(p-1) = l, \text{ si } p - 1 = 7^l \cdot q, \text{ on 7 no divideix a } q.$$

Per tant el subgrup de 7-Sylow $S_7(E(\mathbb{F}_p)) \cong \mathbb{Z}_{7^n} \times \mathbb{Z}_{7^r}$ satisfà que $r \leq l$, on l és la valoració 7-àdica de $p - 1$. La valoració 7-àdica de $p - 1$ també ens permet distingir el cas cíclic del no cíclic tenint en compte els següents aspectes:

- Si $p \equiv 2, 3, 4, 5, 6 (mod\ 7)$, llavors $v_7(p-1) = 0$ i això fa que r també prengui el valor 0 i ens trobem per tant davant del cas cíclic.

- Si $p \equiv 1 (mod\ 7)$, llavors $v_7(p-1) \geq 1$, en aquesta situació, si $r \neq 0$, es tractarà del cas no cíclic en canvi si $r = 0$, es tractarà del cas cíclic.

Conegut el subgrup de 7-Sylow $S_7(E(\mathbb{F}_p)) \cong \mathbb{Z}_{7^n} \times \mathbb{Z}_{7^r}$ d'una corba $E(\mathbb{F}_p)$, el cardinal d'aquesta corba satisfà :

$$\#E(\mathbb{F}_p) = 7^{n+r} \cdot m', \text{ on } m' \text{ no és divisible per 7.}$$

4.3 Punts setena part de punts d'ordre 7^k

Com ja hem explicat en apartats anteriors, a la nostra corba existeixen punts que tenen com a ordre una potència de 7 que denotem com 7^k on $k \geq 1$. Però si el que a nosaltres ens interessa és trobar un punt d'ordre 7^n màxim dins la nostra corba, procedirem tal i com s'explica a [9], donant per suposada l'existència d'un punt d'ordre 7^k que anomenarem P i amb aquest punt mirarem de determinar les condicions necessàries per garantir l'existència d'un punt d'ordre 7^{k+1} que anomenarem Q. Per a que aquest punt $Q = (x, y)$ que busquem d'ordre 7^{k+1} sigui un punt setena part de P, és necessari que es compleixi la següent igualtat:

$$7Q = P$$

Els punts d'ordre 7^{k+1} són, per tant, tots aquells que sumats 7 vegades ens donen un punt d'ordre 7^k. Per trobar aquestos punts en el nostre model de corba, igualem les abscisses dels punts $7Q$ i P, obtenint un polinomi de grau 49. Les arrels d'aquest polinomi es poden trobar a partir de dos de més petits de grau 7 que anomenarem $R_\xi(z)$ i $f_z(x)$ i que mostrem a continuació:

$$
\begin{aligned}
R_\xi(z) = \; & z^7 + 49\xi z^6 + (-7c^7 + 21c^6 - 161c^5 + 196c^4\xi + 280c^4 + 1078c^3\xi \\
& +84c^3 - 588c^2\xi - 224c^2 - 294c\xi + 7c - 196\xi)z^5 + (28c^{10} + 182c^9 + \\
& 294c^8\xi - 1561c^8 - 3038c^7\xi + 3024c^7 + 8869c^6\xi + 2380c^6 - 4312c^5\xi - \\
& 8631c^5 - 7056c^4\xi + 4571c^4 + 1862c^3\xi - 315c^3 + 2695c^2\xi + 322c^2 + \\
& 686c\xi + 294\xi)z^4 + (14c^4 - 420c^{13} - 210c^{12}\xi + 2709c^{12} + 2996c^{11}\xi - \\
& 7574^c11 - 14084c^{10}\xi - 1148c^{10} + 18480c^9\xi + 82320c^9 + 37492c^8\xi - \\
& 201488c^8 - 120596c^7\xi + 190246c^7 + 88200c^6\xi - 75320c^6 - 1596c^5\xi + \\
& 15904c^5 + 210c^4\xi - 5005c^4 - 7196c^3\xi - 42c^3 - 3220c^2\xi - 196c^2 - \\
& 476c\xi - 210\xi)z^3 + (-7c^{17} + 77c^{16}\xi + 168c^{16} - 1344c^{15}\xi + 210c^{15} + \\
& 8043c^{14}\xi + 8785c^{14} - 12544c^{13}\xi - 184072c^{13} - 66542c^{12}\xi + \\
& 966378c^{12} + 356916c^{11}\xi - 2261840c^{11} - 691096c^{10}\xi + 2753366c^{10} + \\
& 562240c^9\xi - 1849743c^9 - 72261c^8\xi + 759822c^8 - 64652c^7\xi - 259868c^7 - \\
& 71393c^6\xi + 69867c^6 + 45276c^5\xi - 4361c^5 + 2107c^4\xi + 1176c^4 + 3528c^3\xi \\
& +42c^3 + 1533c^2\xi + 77c^2 + 112c\xi + 77\xi)z^2 + (-21c^{21} - 14c^{20}\xi + 651c^{20} \\
& +280c^{19}\xi - 7133c^{19} - 1904c^{18}\xi + 22176c^{18} + 2156c^{17}\xi + 153545c^{17} + \\
& 40026c^{16}\xi - 1613164c^{16} - 249312c^{15}\xi + 6338395c^{15} + 645106c^{14} \\
& \xi - 13787480c^{14} - 600698c^{13}\xi + 18231374c^{13} - 852852c^{12}\xi - \\
& 15347997c^{12} + 2967860c^{11}\xi + 8828575c^{11} - 3218320c^{10}\xi - 4043186c^{10} \\
& +1373344c^9\xi + 1616384c^9 + 22736c^8\xi - 463393c^8 - 151424c^7\xi + \\
& 88907c^7 + 42378c^6\xi - 15995c^6 - 18578c^5\xi - 1547c^5 - 168c^4\xi - 112c^4 - \\
& 294c^3\xi + 35c^3 - 322c^2\xi - 14c^2 - 14\xi)z^1 + (c^{25} + c^{24}\xi + 10c^{24} - 22c^{23}\xi \\
& -1308c^{23} + 159c^{22}\xi + 27314c^{22} - 20c^{21}\xi - 284089c^{21} - 6363c^{20}\xi + \\
& 1776838c^{20} + 38710c^{19}\xi - 7218071c^{19} - 69153c^{18}\xi + 19803153c^{18} - \\
& 303682c^{17}\xi - 37510728c^{17} + 2313123c^{16}\xi + 49810389c^{16} - \\
& 7303966c^{15}\xi - 47205305c^{15} + 13901159c^{14}\xi + 33062403c^{14} - \\
& 16924180c^{13}\xi - 18263028c^{13} + 12983481c^{12}\xi + 8521030c^{12} - \\
& 5839092c^{11}\xi - 3397781c^{11} + 1258192c^{10}\xi + 1155715c^{10} + 4594c^9\xi \\
& -336072c^9 - 84577c^8\xi + 62482c^8 + 38092c^7\xi - 4666c^7 - 8596c^6\xi \\
& +1092c^6 + 2058c^5\xi + 2058c^5\xi + 560c^5 + 91c^4\xi + 64c^4 - 34c^3\xi - 4c^3 \\
& +27c^2\xi + c^2 - 2c\xi + \xi).
\end{aligned}
$$

$$
\begin{aligned}
f_z(x) = \; & x^7 + zx^6 + (c7 - 3c^6 - 5c^5 + 16c^4 - 2c^3z - 12c^3 + 4c^2 + 2cz - c)x^5 + \\
& (-4c^9 + 25c^8 - 50c^7 + c^6z + 46c^6 + 2c^5z - 23c^5 - 6c^4z + 7c^4 + 2c^3z \\
& -c^3 + c^2z)x^4 + (c^{12} - 6c^{11} + 12c^{10} - 14c^9 - 2c^8z + 12c^8 + 4c^7z - \\
& 4c^7 - 2c^6 - 4c^5z + 2c^4z + c^4)x^3 + (-c^{13} + 6c^{12} - 8c^{11} + c^{10}z - \\
& 5c^{10} - 4c^9z + 15c^9 + 6c^8z - 4c^7z - 6c^7 + c^6z + 3c^6)x^2 + (c^{15} - \\
& 8c^{14} + 22c^{13} - 25c^{12} + 5c^{11} + 14c^{10} - 12c^9 + 3c^8)x^1 + (c^{16} - \\
& 6c^{15} + 15c^{14} - 20c^{13} + 15c^{12} - 6c^{11} + c^{10}).
\end{aligned}
$$

Per trobar aquestos punts haurem de seguir els passos que mostrem a continuació. Busquem les arrels del polinomi $R_\xi(z)$:

- Si no trobem cap arrel, ja no cal mirar res més, doncs aquest fet ens indica que no existeix cap punt Q de la corba, que tingui ordre 7^{k+1} tal que $7Q = P$, és a dir, que punt P ja és un punt d'ordre 7^k màxim, i per tant, la cerca haurà acabat.

- En cas contrari, si trobem solucions a $R_\xi(z)$ les guardem dintre un vector i li anem passant al polinomi $f_z(x)$ podent-nos trobar novament, amb dos situacions :
  - Si amb cap arrel obtinguda a $R_\xi(z)$ obtenim solució a $f_z(x)$ significa que no existeix cap punt Q de la corba, amb ordre 7^{k+1} tal que $7Q = P$ i per tant ja no cal seguir.
  - Si trobem solucions en algun $f_z(x)$, exactament 7 solucions, això ens indicarà que hi ha punts d'ordre 7^{k+1}.

4.4 Obtenció dels valors n i r del subgrup de 7-Sylow

En aquest capítol veurem com calcular el subgrup de 7-Sylow mitjançant l'obtenció dels valors n i r. Aquest procediment variarà una mica, depenent de si ens trobem en el cas cíclic o en el no cíclic.

Seguidament explicarem com actuarem davant de cadascun dels casos.

4.4.1 Càlcul del subgrup 7-Sylow per al cas cíclic

El procediment per trobar el valor de la n es basa en el procés descrit a l'apartat anterior per trobar els punts d'ordre 7^{k+1}, és a dir, fent ús dels polinomis $R_\xi(z)$ i $f_z(x)$. Primer mirem si hi ha solucions a $R_\xi(z)$ amb el valor inicial de $\xi = 0$, que correspon al punt $P = (0, 0)$. Si en trobem, mirarem si n'hi ha a $f_z(x)$ i cada cop que trobem solucions als dos polinomis, és a dir, cada cop que trobem un punt d'ordre 7^{k+1} la nostra n anirà augmentant. Si tractem de representar amb un arbre els progressos de la n, obtindrem un arbre de punts com el que mostra la següent figura:

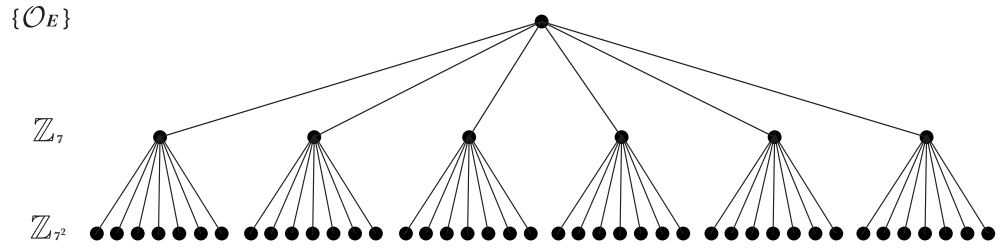

Figura 4.1: Arbre en el cas cíclic ($S_7(E(\mathbb{F}_p)) \cong \mathbb{Z}_{7^2}$)

Notem que l'algorisme comença en el nivell 1 amb el punt $P = (0, 0)$ i baixa per un camí descendent fins arribar al nivell màxim, que serà n.
En efecte, com que ens trobem en el cas en que el subgrup de 7-Sylow és cíclic, l'arbre format pels punts d'aquest subgrup serà complet excepte en el nivell 0. Aquesta dada ens és de gran ajuda ja que si en un nivell un punt té un punt setena part, tots els punts del mateix nivell també en tindran, és a dir que sabent això, només ens cal comprovar l'existència d'un únic punt setena part per a cada nivell.

4.4.2 Càlcul del subgrup 7-Sylow per al cas no cíclic

De la mateixa manera que al cas cíclic, el procés que realitzarem és comprovar l'existència de punts d'ordre 7^{k+1} a partir d'un punt d'ordre 7^k. El procés en el cas no cíclic serà similar al del cas cíclic, amb la diferencia que aquest cop el número de punts d'ordre 7 que obtenim amb el polinomi de 7-divisió és de 48. Si al cas cíclic l'arbre que hem obtingut tenia les branques equilibrades, en aquest cas n'hi poden haver de diferents longituds (vegis figura 4.2 on en el nivell 1 de l'arbre s'han dibuixat 2 punts generadors de $E[7](\mathbb{F}_p)$ en comptes dels 48 punts.

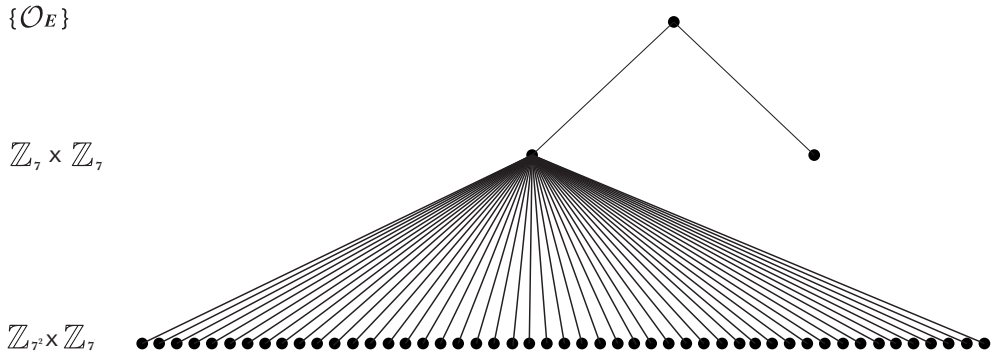

Figura 4.2: Arbre en el cas no cíclic ($S_7(E(\mathbb{F}_p)) \cong \mathbb{Z}_{7^2} \times \mathbb{Z}_7$)

El procés que haurem de seguir per obtenir la n serà doncs el següent:

- Agafem dos arrels del polinomi de 7 divisió, una serà $\xi = 0$ i l'altra $\xi = segchi$, de manera que ξ no sigui abscissa de cap dels múltiples de $P = (0, 0)$. D'aquesta manera l'altra abscissa correspon a l'altre generador del grup $E[7](\mathbb{F}_p)$

- Un cop fet això, hem d'obtenir la n en el cas en que $\xi = 0$ i en el cas en el que la $\xi = segchi$ tal i com s'ha fet en el cas cíclic.

- Hem obtingut doncs 2 valors, i el que hem de fer ara és mirar quins d'aquests valors és més gran per tal de saber per quina branca hem baixat més. Al més gran l'anomenarem n i al més petit r, en el cas en que n i r fossin iguals actuaríem de la mateixa manera. Busquem mitjançant l'equació de la corba, una ordenada per cadascuna de les abscisses obtingudes(una corresponent al valor n i l'altra al valor r). Denotarem per P i Q els punts d'ordre 7^n i 7^r, respectivament. Però encara no hem acabat, doncs el valor de n encara pot augmentar i a nosaltres ens interessa trobar el valor màxim que pot obtindre.

- Per mirar si encara podem baixar més, hem de sumar aquests 2 primers punts, P i Q que acabem de trobar. D'aquesta manera trobarem un punt d'ordre 7^n al que anomenarem P', ja que serà el nostre nou punt P, situat en una altra branca diferent per la qual mirarem de baixar.

- Podem trobar-nos amb dos situacions:
  - Si podem baixar amb P' tornarem a començar el procés com si fos la nostra nova P i mirarem de continuar baixant fins que puguem.
  - En el cas de que no puguem baixar, el que hem de fer és anar sumant a P' el valor de Q, si no podem baixar, li sumarem $2Q$ i així anirem sumant Q fins arribar a un màxim de $6Q$. Si arribem a $6Q$ i no hem pogut baixar, el valor que en aquell moment tingui la n serà el valor de n màxim que buscàvem. Si per el contrari, en algun moment podem baixar més, tornarem a començar el procés.

Capítol 5

Implementació

En aquest capítol presentem el software i el hardware que hem utilitzat per realitzar aquest projecte tant la part pràctica del propi algoritme com la part de redacció del projecte. També mostrem la implementació de l'algorisme per determinar el subgrup de 7-Sylow d'una corba el·líptica.

5.1 Software i Hardware utilitzat

El software del qual hem fet ús en aquest projecte és el següent:

- Part pràctica:
  - Sistema operatiu Linux Fedora Core 4, Kernel 2.6.11-1.1369
  - Editor de text *gedit*2.8.1
  - Compilador gnu/g++ versió 3.4.2
  - Llibreria matemàtica LiDIA versió 2.1 (Maig 2001)
- Part de redacció:
  - Sistema operatiu, MAC OSX 10.4.10
  - Editor i compilador, TeXShop Versió 2 (2.09)
  - Sistema d'edició de documents LaTeX(teTeX)

El hardware que hem utilitzat en aquest projecte és el següent:

- Part pràctica:
  - Pentium IV
  - Velocitat de la CPU 1.7Ghz
  - 1024 MB de memòria RAM
- Part de redacció:
  - Intel Core 2 Duo
  - Velocitat de la CPU 2Ghz

- 2048 MB de memòria RAM

Tota la part pràctica del treball ha estat feta mitjançant el llenguatge de programació d'alt nivell C++, ja que la llibreria matemàtica LiDIA que és la que nosaltres utilitzem està basada en aquest llenguatge de programació.

5.2 Petita introducció a la llibreria LiDIA

Per a poder fer la part pràctica del nostre treball, era necessari treballar amb una llibreria matemàtica d'alt nivell que C++ no inclou per defecte. LiDIA és una llibreria per a la computació numèrica, de la Universitat de Tecnologia de Darmstadt, segons la descripció original: "LiDIA is a C++ library for computational number theory which provides a collection of hightly optimized implementations of various multiprecision data types and time-intensive algorithms". La seva distribució inclou el codi font i és de lliure distribució per usos no comercials.
LiDIA ve recopilada en diferentes llibreries que contenen diverses classes, aquestes són les classes que nosaltres hem utilitzat:

- ***bigint:*** Ens proporciona tota l'aritmètica necessària per treballar amb enters grans. Les accions i funcions més destacades d'aquesta classe són:
  - ***bigint next_prime(const bigint &a)***
    Retorna el següent nombre primer que existeix després de $\boldsymbol{a}$.
  - ***bigint randomize(const bigint &a)***
    Retorna un nombre a l'atzar $\boldsymbol{x} \in$ [0,...,$\boldsymbol{a}$-1] si $\boldsymbol{a} > 0$, $\boldsymbol{x} \in$ [$\boldsymbol{a}$+1,...,0] si $\boldsymbol{a} < 0$ i altrament llença una excepció.
  - ***int string_to_bigint(const char *s, bigint &a)***
    Converteix la variable $\boldsymbol{s}$ a bigint i la guarda a $\boldsymbol{a}$ retornant el nombre de caràcters utilitzats per fer la conversió.
- ***galois_field:*** Ens permet definir una variable que crea un cos fintit $\mathbb{F}_p$
- ***gf_element:*** Ens permet representar elements de la classe ***galois_field***, així com també fer operacions sobre ells. Les accions i funcions més destacades d'aquesta classe són:
  - ***bigint e.lift_to_Z()***
    Si $\boldsymbol{e}$ és un element d'un cos primer $\mathbb{F}_p$ o d'un anell $\mathbb{Z}_n$, llavors, la funció retorna l'últim residu no negatiu de la classe $\boldsymbol{e}$ mòdul p.
  - ***Fp_polynomial & e.polynomial_rep()***
    Retorna la representació polinòmica de l'element $\boldsymbol{e}$.
- ***elliptic_curve:*** Ens permet representar i treballar amb corbes el·líptiques.
- ***point:*** Ens permet definir i realitzar les operacions bàsiques sobre punts d'una corba el·líptica.
- ***polynomial:*** Ens permet treballar sobre polinomis.

- ***Fp_polynomial:*** Ens permet treballar amb un polinomi mòdul un nombre enter. La funció més destacada d'aquesta classe és:
  - ***base_vector<bigint> find_roots (const Fp_polynomial &f, int flag = 0)***
    Retorna la llista d'arrels de $\boldsymbol{f}$. Si ***flag*** $\neq 0$, $\boldsymbol{f}$ ha de ser mònic i el producte del grau de $\boldsymbol{f}$ serà diferent a les arrels, sinó no es fa cap suposició sobre $\boldsymbol{f}$. Per calcular-les utilitza l'algoritme probabilístic de Berlekamp.
- ***base_vector:*** Ens permet crear vectors implementant les operacions bàsiques d'accés.
- ***timer:*** Ens permet tenir un control sobre el temps.

5.3 Algorismes

En aquest apartat explicarem els algorismes que hem utilitzat per trobar el subgrup de 7-Sylow d'una corba el·líptica $E/\mathbb{F}_p$. L'algorisme principal és el que hem anomenat (Sylow). S'encarrega de comprovar que el discriminant no sigui 0, i de mirar si la p que introduïm és congruent, o no, a 1 mòdul 7, i diferenciar per tant, si ens trobem en el cas cíclic o en el no cíclic. En funció de quin sigui el cas en el que ens trobem cridarà a altres algorismes que permetran trobar els valors de n i r.

A continuació explicarem detalladament cadascun dels algorismes que hem utilitzat, emprant una espècie de pseudocodi. Abans però, definirem les variables globals necessàries per poder ententre millor cada algorisme.

Variables globals:

chi1, p, p1, c1 : [**Enter gran**];
c, c2, ..., c25 : [**Elements del cos**];
f, fz : [**Polinomi d'elements del cos**];

Algorisme 1 *(Determinació del subgrup de 7-Sylow de $E/\mathbb{F}_p$)*

Entrada: Un primer senar p i la variable c que ens generarà la corba el·líptica $E/\mathbb{F}_p$ d'equació $y^2 + a_1xy + a_3y = x^3 + a_2x^2$.

Sortida: Dos enters n i r tals que $S_7(E/\mathbb{F}_p) \cong \mathbb{Z}_{7^n} \times \mathbb{Z}_{7^r}$, una abscissa ξ_n d'un punt d'ordre 7^n i una abscissa ξ_r d'un punt d'ordre 7^r en el cas que existeixi i 0 en el cas que no existeixi.

algorisme Sylow(ps(E/S), cs(E/S), n(E/S), P(E/S), r(E/S), Q(E/S))
p:=ps;
c:=cs;
segchi:=0 [**Enter gran**];

vqsol : [**Boleà**];
c2d : [**Element del cos**] ;
c3d : [**Enter gran**] ;
c3d : = c2d; [**Conversió a enter de c2d**]

si $p \equiv 1 \pmod 7$ **llavors** [**CONGRUENT**]
segchi:=$Div\_7(p, cd3)$;
si segchi $\neq$ -1 **llavors** [**CAS NO CÍCLIC**]
vqsol:=cert;
sinó [**CAS CÍCLIC**]
vqsol:=fals;
fsi
sinó si $p \pmod 7 \neq 1$ **llavors** [**NO CONGRUENT**]
vqsol:=fals;
fsi

g1(p) : [**Cos g1(p)**];
discr:=0 [**Elements del cos**] ;
discr:=$1 + 5 * c - 8(c * c) + (c * c * c)$;

si discr=0 **llavors**
[**Discriminant=0, no seguim**] ;
sinó
[**Inicialització de totes les c**];
f:=**(Mòdul p)**;
fz:=**(Mòdul p)**;
x(F): [**Element del cos**] ; [**Abscissa del punt P**]
u(F): [**Element del cos**] ; [**Abscissa del punt Q**]
i, k : [**Enter gran**];
E : [**Corba el-líptica de gf_elements que depen de c**];
P :=0;
Q :=0;
si vqsol=fals **llavors** [**CAS CÍCLIC**]
n :=1;
baixar(x,n);
P(abscissa):=x;
sinó [**CAS NO CÍCLIC**]
P(abscissa):=x;
u:=segchi;

```
        n:=1;
        baixar(x,n);
        r:=1;
        baixar(u,r);
        si n < r llavors
          swap(n,r); [Intercanviem n per r]
          swap(x,u); [Intercanviem x per u]
        fsi
          Q(abscissa):=u;
          fer
            i:=0;
            k:=n;
            P(abscissa):=x;
            fer
              i:=i+1;
              x:=(P+i·Q);
              baixar(x,n);
            fmentres n=k i i < 6
          fmentres n ≠ k
      fsi (Final cas no cíclic)
    fsi (Final discriminant != 0)
  Falgorisme
```

El segon algorisme al que hem anomenat **baixar** s'encarrega de mirar de baixar el màxim possible i obtenir finalment el nivell de descens.

Algorisme 2 *(Descens subgrup 7-Sylow)*

Entrada: Una variable de tipus gf_element **chi** d'entrada-sortida i una variable enter gran també d'entrada-sortida amb la qual portarem el compte dels nivells que anem baixant a l'arbre.
Sortida: La sortida són les mateixes variables que a la entrada amb les modificacions que hagin rebut a la la funció.

```
algorisme baixar(chi(E/S), niv(E/S))

  aux:=chi.get_field [Element del cos];
  b : [Boleà];
  sz1, sz2, i : [Lidia_tamany_t] ;
  bv, bv2 : [Vector d'enters grans] ;

  fer                          [Si chi i fz tenen solucions BAIXEM]
```

b:=fals;
i:=-1;
polchi(chi); **[Cridem a la funció del polinomi chi]**
bv:=find_roots(f); **[Trobem les arrels de la funció chi]**
sz1:= bv.size()-1; **[Nombre d'arrels trobades a f]**
mentres $b = fals$ i $i < sz1$ **fer**
i:=i+1;
aux:=bv[i];
polfz(aux); **[Crida la funció polfz amb aux]**
bv2:=$find\_roots(fz)$; **[Trobem les arrels de la funció fz]**
sz2:=bv2.size(); **[Nombre d'arrels trobades a f]**
si sz2 $\neq$ 0 **llavors**
b:=cert;
fsi
fmentres
si b:=cert **llavors**
niv:=niv+1; **[Baixem un nivell de l'arbre]**
chi:=bv2[1]; **[Chi pren el valor de la primera arrel de bv2]**
fsi
ffer mentres b=fals
Falgorisme

El tercer algorisme, **Div_7**, s'encarrega de mitjançant el polinomi de 7 divisió obtenir una solució la qual no pot ser abscissa de cap dels múltiples de $P = (0, 0)$.

Algorisme 3 *(Polinomi de 7-divisió)*

Entrada: Dos variables del tipus **enter gran**, la primera d'elles **p1** serà un primer i **c1** serà un nombre qualsevol més gran que 2.
Sortida: La sortida és un enter que contindrà un punt, **puntxdif**, solució del polinomi de 7-divisió, el qual no pot ser abscissa de cap dels múltiples de $P = (0, 0)$.

algorisme Div_7(p1,c1)

p, c, size7div : **[Enter gran]**;
p:=p1 ;
c:=c1 ;
Fp(p): **[Cos $\mathbb{F}_p$]**;
a1, a2, a3, a4, a6(Fp): **[Element del cos]**;
[Inicialitzem a1, a2, a3, a4 i a6 per poder definir l'equació de la corba] ;
[Declaració de la corba E] ;

E.get_bi(b2, b4, b6, b8): [**Element del cos**];
f: [**Cos** $\mathbb{F}_p$] ;
[**Assignem a f el polinomi de 7 divisió**] ;
bv: [**Vector d'enters grans**] ;
bv:= [**Vector amb les Arrels trobades de f**] ;
size7div:= [**Nombre d'arrels trobades**] ;

si nombre d'arrels = 24 **llavors**
[Declarem 3 enters grans on guardarem les abscisses de 3 punts]
[Declarem una variable enter gran **puntxdif** on guardarem un punt obtingut a partir de les arrels obtingudes de f que serà diferent als 3 punts anteriorment obtinguts]
cont:=0 [**Enter**]
chin: [**Element del cos**]
chin:=0
punt: **Declarem un punt de la corba E i li asignem l'abscissa de chin**
(Guardem les abscisses dels punts P=(0,0), 2P i 3P de la corba, en les variables anteriorment declarades)
mentres no trobem una arrel (de les 24 obtingudes) diferent a les abscisses dels punts que hem calculat i guardat **llavors**
cont:=cont+1; [**Passem a la següent arrel**]
fmentres
retorna puntxdif; [**Arrel diferent de P=(0,0), 2P, 3P**]
sinó si nombre d'arrels = 3 **llavors**
retorna -1;
fsi

Falgorisme

Capítol 6

Resultats i conclusions

En aquest capítol mostrarem les proves que hem realitzat fent ús dels algorismes que hem implementat i els resultats que hem obtingut en les mateixes. I a l'últim apartat exposarem les conclusions a les que s'han arribat a partir d'aquestes proves i comentarem les línies de treball que seria convenient estudiar properament.

6.1 Resultats

A les dos taules següents mostrem els resultats del 7-Sylow per a cadascuna de les corbes generades al donar tots els possibles valors a c, és a dir, $2 \leq c \leq 701$ el cas en el que la $p = 701$ i $2 \leq c \leq 7001$ en el cas que $p = 7001$. Aquests dos nombres primers, són congruents a 1 mòdul 7 i per tant la r pot ser diferent de 0.

$p = 701$

$n + r$	n	r	Nombre corbes	Total corbes
1	1	0	522	522
2	1	1	48	
2	2	0	42	90
3	2	1	48	
3	3	0	36	84

Taula 6.1: Distribució de les corbes sobre $\mathbb{F}_p$ segons el 7-Sylow, amb p=701

La taula 6.1 ens mostra els valors de la n i de la r i el nombre de corbes que obtenim sobre el cos $\mathbb{F}_{701}$, que tenen el mateix 7-Sylow. Si observem la taula ens adonarem de que si sumem la quantitat de corbes obtingudes no ens dóna 701, sinó 696, les 5 corbes que falten tenen discriminant 0.

La taula 6.2 tal com passava amb $p = 701$, arriba al tercer nivell. Si observem tan la taula 6.2 com l'anterior, ens adonem d'un detall, al mateix nivell, en els casos en els quals la r pren algun valor diferent a 0, el nombre de corbes és major que en els cas que aquesta és 0.

p=7001

$n+r$	n	r	Nombre corbes	Total corbes
1	1	0	4980	4980
2	1	1	924	
2	2	0	840	1764
3	2	1	144	
3	3	0	108	252

Taula 6.2: Distribució de les corbes $\mathbb{F}_p$ segons el 7-Sylow, amb p=7001

Primer p	Cong (mod 7)	Nombre Corbes	T.exec. (seg)	T.corba (seg)
263	4	261	0,5	0,00191
661	3	659	1,38	0,00149
2221	2	2219	4,52	0,00203
239	1	234	1,17	0,005
659	1	654	3,34	0,0051
2269	1	2266	12,40	0,0054
$2 * 10^9 + 203$	5	1000	3,14	0,00314
$10^{60} + 3201$	2	1000	85,51	0,08551
$10^{100} + 267$	$\neq 1$	1000	247,93	0,24793

Taula 6.3: Taula de temps del càlcul de 7-Sylow

A la taula 6.3 mostrem el temps d'execució i el temps mig per corba que ha sigut necessari per calcular el 7-Sylow de corbes per a cada cos $\mathbb{F}_p$ relacionat a la taula. Per poder apreciar la diferència de temps, entre quan és congruent a 1 mòdul 7 i quan no ho és, hem agafat en els 6 primers casos de la taula, 3 primers no congruents a 7 mòdul 1 i 3 més, de congruents, sent el marge entre ells molt petit. Si observem doncs, la penúltima columna de la taula, veiem que el temps d'execució és clarament superior en el cas en que el primer és congruent a 1 mòdul 7, sent la diferència de més del doble. Això és degut a que en el cas congruent, hi trobarem corbes amb 7-Sylow no cíclic, on el procediment per trobar la n i la r és més complex i conseqüentment, molt més lent.

Primer p	Cong (mod 7)	c	n	r	temps(seg.)
100dcong	1	1775	5	1	7.46

Taula 6.4: 7-Sylow amb un nombre de nivells elevat

A la taula 6.4 hem mirat de buscar valors de n elevats, i ho hem fet en el cas d'un primer molt gran, exactament de 100 dígits que a més a més és congruent

a 1 mòdul 7, amb lo qual r pot prendre valors diferents a 0. El primer és:

$$721261014729547490954452378504349240996938214818676546008250008539351955652592145558870542302075142 1.$$

Al ser tan gran a la taula l'hem anomenat **100dcong**. Amb aquest mateix nombre, s'han mirat les 1000 primeres corbes i el temps emprat ha estat de **1461,44** segons on ens queda per tant, un temps mig per corba de **1,461** segons.

A continuació veurem 2 captures de l'obtenció dels resultats que queden reflectits a les taules a partir de les diferents variants de l'algorisme que hem desenvolupat.

Versió 1
Amb aquesta primera variant l'usuari introdueix per teclat un primer p que definirà el cos i una única c que definirà la corba.

```
[v4769079@localhost Desktop]$ g++ -O 7sylowa.cc -I /usr/local/include -L /usr/lo
cal/lib -o 7sylowa -lLiDIA -lgmp -lm
[v4769079@localhost Desktop]$ ./7sylowa
Inici del Programa -----
Introduim la P
239
Introduim la C
13
cas congruent
ARRELS del polinomi de 7 divisio----[ 114 168 77 92 53 180 156 41 34 204 182 126
 227 149 218 129 103 112 165 116 209 183 30 0 ]
114
cas no ciclic
La n i la r obtingudes (n,r) son :(1,1)
Final del Programa -----
El temps d'execucio es:0.02
[v4769079@localhost Desktop]$
```

Figura 6.1: Càlcul de n i r amb la versió 1 de l'algorisme de 7-Sylow per $p = 239$ i $c = 13$

Versió 2
En aquesta segona captura hem fet ús de la segona variant del nostre algorisme el qual a partir d'un primer p que introduïm per teclat, trobarà la n i la r de totes les corbes del cos $\mathbb{F}_p$.

```
cas congruent
ARRELS del polinomi de 7 divisio----[ 0 203 12 ]
ARRELS del polinomi de 7 divisio----[ 0 203 12 ]
cas ciclic
(1,0)
Final del Programa -----
la següent c es(237)
cas congruent
ARRELS del polinomi de 7 divisio----[ 0 227 6 ]
ARRELS del polinomi de 7 divisio----[ 0 227 6 ]
cas ciclic
(1,0)
Final del Programa -----
la següent c es(238)
cas congruent
ARRELS del polinomi de 7 divisio----[ 0 2 237 ]
ARRELS del polinomi de 7 divisio----[ 0 2 237 ]
cas ciclic
(1,0)
Final del Programa -----
la següent c es(0)
r=0 n=0 ---3
r=1 n=0 ---0
r=0 n=1 ---192
r=1 n=1 ---24
r=2 n=0 ---0
r=0 n=2 ---18
r=2 n=1 ---0
r=3 n=0 ---0
r=1 n=2 ---0
r=0 n=3 ---0
El temps d'execucio es:1.13
[v4769079@localhost Alg usados final]$
```

Figura 6.2: Càlcul de la n i la r de totes les corbes del cos $\mathbb{F}_p$ per p=239

6.2 Conclusions i futures linies de treball

En aquesta secció parlarem dels objectius aconseguits durant la realització d'aquest treball, i comentarem quines són les possibles línies de treball que seria convenient seguir.

6.2.1 Conclusions

En aquest treball s'ha desenvolupat i implementat un algorisme que ens permet trobar el 7-Sylow d'una corba el·líptica sobre un cos $\mathbb{F}_p$, utilitzant com a paràmetres d'entrada, un primer p i els coeficients d'una corba el·líptica sobre $\mathbb{F}_p$. Aquest és un cas particular de l'algoritme donat a [9] per determinar el 7-Sylow d'una corba el·líptica.
Amb l'algoritme inicial s'han elaborat dos variants per poder realitzar les probes oportunes:

- La primera variant de l'algorisme ens ha servit per trobar el Sylow d'una corba determinada. A les probes s'han introduït tot tipus de nombres primers tant congruents com no congruents a 1 mòdul 7 i també nombres primers de fins a 100 digits, per tal d'observar el comportament de

l'algorisme en cada cas.

- La segona variant de l'algoritme s'encarrega de trobar el Sylow d'un rang de corbes sobre un cos $\mathbb{F}_p$. A les probes s'ha comprovat el nombre de corbes vàlides i el temps que ha set necessari per obtenir tots els Sylows

6.2.2 Futures línies de treball

Com a futura línia de treball més immediata, seria la de l'estudi de la generació de volcans de 7-isogènies, ja que es pot utilitzar la informació obtinguda sobre l'estructura del subgrup de 7-Sylow. L'algorisme a desenvolupar, s'haurà d'encarregar de a partir d'una corba trobar un camí fins al cràter del volcà al qual pertany.

Una altra línia de treball a considerar seria el completar altres ℓ-Sylows i altres tècniques per mirar d'arribar a obtenir el cardinal d'una corba el·líptica sobre un cos finit, el qual és un problema computacionalment difícil.

Bibliografia

[1] A. Albajes. *Corbes el·líptiques: Un criptosistema semànticament segur i determinació de la 2^n-torsió.* Treball de Fi de Carrera, Universitat de Lleida, Gener 2003.

[2] J. Barrientos, T. Bautista, T.Oetiker, H. Partl, I. Hyna, i E. Schlegl. *Una Descripció de LaTeX 2_ε.* Versió 0.3, 25 de febrer de 2003.

[3] J. Gimbert i J. M. Miret. *Problemes d'Algebra per a ciències de la computació.* Edicions de la Universitat de Lleida, 1997.

[4] Lidia Group. *LiDIA, a library for computational number theory. Reference Manual.* http://www.informatik.tudarmastadt.de/TI/LiDIA, Edition 2.1.1, May 2004.

[5] N. Koblitz. *Elliptic curve cryptosystems. Maths of Computation,* 48, 1987.

[6] V. Miller. *Use of elliptic curves in cryptography. Advances in Cryptology*, 1986.

[7] J. Molgó. *Determinació del subgrup de l^n-torsió de les corbes el·líptiques definides sobre un cos finit* Treball de Final de Carrera, Universitat de Lleida, Febrer de 2004.

[8] O. Morelló i S. Serramona. *Determinació del subgrup de 5-Sylow d'una corba el·líptica i generació de volcans de 5-isogènies.* Treball de Fi de Carrera, Universitat de Lleida, Octubre 2006.

[9] R. Moreno. *Grupos de torsión de las curvas elípticas definidas sobre cuerpos finitos.* PhD thesis, Universitat Politècnica de Catalunya, 2005.

[10] D. Pardell i J. Valera.*Determinació de la 3^n-torsió d'una corba el·líptica i criptosistema de Meyer i Müller basat en punts de tercera part.* Treball de Fi de Carrera, Universitat de Lleida, Setembre 2003.

[11] E. Porta. *Algorisme per determinar la 3^n-torsió d'una corba el·líptica.* Treball de Fi de Carrera, Universitat de Lleida, Setembre 2003.

[12] R. Schoof. *Elliptic curves over finite fields and the computation of square roots mod p.* Mathematics of Computation, vol. 44, pages 483-494, 1985.

[13] D. Shanks. *Class number, a theory of factorization an genera*. Proc. Symposium of Pure Math. 20, Amer. Math. Soc., pages 415-440, 1970. in Mathematics (106). Springer-Verlag, 1986.

[14] J. H. Silverman. *The Arithmetic of Elliptic Curves.* Graduate Texts